Optimal Fuzzy Rules Cover Extrema

Bart Kosko
Department of Electrical Engineering, Signal and Image Processing Institute, University of Southern California, Los Angeles, California 90089-2564

A fuzzy system approximates a function by covering the graph of the function with fuzzy rule patches and averaging patches that overlap. But the number of rules grows exponentially with the total number of input and output variables. The best rules cover the extrema or bumps in the function—they *patch the bumps*. For mean-squared approximation this follows from the mean value theorem of calculus. Optimal rules can help reduce the computational burden. To find them we can find or learn the zeroes of the derivative map and then center input fuzzy sets at these points. Neural systems can then both tune these rules and add rules to improve the function approximation. © 1995 John Wiley & Sons, Inc.

I. FUZZY FUNCTION APPROXIMATION AND THE CURSE OF DIMENSIONALITY

A fuzzy system needs too many fuzzy rules to approximate most functions. The number of rules grows exponentially with the number of input and output variables. In the end this ''curse of dimensionality'' can defeat an expert who guesses at the rules or a neural system that tries to learn the rules from data.

The rule geometry shows the problem. The rules define fuzzy patches that can cover part of the graph of the function. An additive fuzzy system adds or averages patches that overlap and can always approximate a continuous function on a compact set with a finite number of rules.[1] For $f\colon R \to R$ it takes k rule patches in the plane to cover the graph. For $f\colon R^2 \to R$ it takes on the order of k^2 rules to cover the surface in some 3-D rectangle. In general for $f\colon R^n \to R^p$ it takes on the order of k^{n+p-1} rules to cover the graph of f.

Optimal rules can reduce the number of rules used to approximate a function. Neural learning tends to find some of these rules[2,3] and so can prune the rule set as well as tune it. In theory we can find the best rules by minimizing the mean-squared error of the approximation for a given fuzzy architecture. A complete closed-form solution depends on the shape of the fuzzy sets and how the system converts inputs to outputs.

CCC 0884-8173/95/020249-07

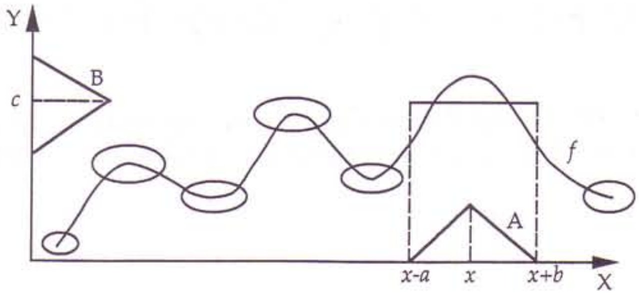

Figure 1. Optimal fuzzy rule patches cover the extrema of a function. A lone rule defines a flat line segment that cuts the graph of the local extremum in at least two places. The mean value theorem implies that the extremum lies between these points.

A natural place to put the rule patches is at the extrema or bumps of the function (including its endpoints). We show that this is the best place in the mean-squared sense. Figure 1 shows how the rules might patch the bumps in a smooth function.

This result gives a new way to approximate $f: X \to Y$. First find the zeroes of the derivative map f'. Neural or direct methods can estimate f' from the difference of noisy samples $(x, f(x))$. Then Newton's method[4] or other iterative or contraction maps can find some or all of the root values $\hat{x}$ such that $f'(\hat{x}) = 0$. Then center the input fuzzy sets at these roots and perhaps add fuzzy sets centered between the roots. Or clustering algorithms can estimate the bumps directly. Supervised or unsupervised learning can further tune the rules.

II. FUZZY SYSTEMS WITH ONE FUZZY RULE

A fuzzy system is a set of fuzzy rules that maps inputs to outputs. So a fuzzy system F is a map $F: X \to Y$.

We want F to approximate some function $f: X \to Y$ in the sense of least squares.[5] We want to minimize the total squared error E over all x in X:

$$E = \int_X (f(x) - F(x))^2 \, dx \tag{1}$$

$$= \int_v^u (f(x) - F(x))^2 \, dx \tag{2}$$

when $X = [u, v]$ for real constants u and v. We will work with the scalar case for simplicity.

Consider first the minimal fuzzy system with just one fuzzy rule of the form "If X is A, then Y is B" for fuzzy sets A and B. Suppose A is nonzero on some subinterval $[x - a, x + b]$ of $[u, v]$ and zero elsewhere for some x in $[u, v]$ and for constants a and b. The constants a and b are non-negative. For each x either $a > 0$ or $b > 0$ or both. The cases $a = 0$ and $b = 0$ deal with endpoint extrema.

The shape of fuzzy set A does not matter. We just need $m_A(z) > 0$ for z in the subinterval and $m_A(z) = 0$ on its complement in $[u, v]$. Here m_A: $[u, v] \rightarrow [0, 1]$ is the indicator function or set of fuzzy set A.

Suppose fuzzy set B is not empty. So $m_B(y) > 0$ for at least one y in the real output space Y. The shape of B does not matter for correlation-product inference[6] or "scaling." B must be symmetric for correlation-minimum inference of "min clipping" since then B has the same centroid as $B' = \min(m_A(z), B)$ for all z in the subinterval. In practice B is connected. It need not be. But then we could view the rule "If X is A, then Y is B" as two or more rules of the form "If X is A, then Y is B_1" and "If X is A, then Y is B_2" where B_1 and B_2 are two of the disjoint components of B. So assume B is connected. Then the rule patch $A \times B$ is connected and a patch proper.

A fuzzy system F with one rule defines a rectangular pulse. $F(z) = \text{Centroid}(B)$ for all z in the subinterval $[x - a, x + b]$. Else $F(z) = 0$. This holds for both centroid and mode defuzzification. Consider first centroid defuzzification. Then $y = F(x) = \text{Centroid}(B') = \text{Centroid}(m_A(z)\, B) = \text{Centroid}(B)$ on the subinterval since

$$F(z) = \frac{\int_Y m_A(z)\, m_B(y)\, y\, dy}{\int_Y m_A(z)\, m_B(y)\, dy} \tag{3}$$

$$= \frac{m_A(z) \int_Y m_B(y)\, y\, dy}{m_A(z) \int_Y m_B(y)\, dy} \tag{4}$$

$$= \text{Centroid}(B) \tag{5}$$

for all z such that $m_A(z) > 0$.

For mode defuzzification $y = F(z)$ if $m_B(y) = \sup m_B(w)$ where the supremum ranges over all w in the output space Y. But $\sup m_A(z)\, m_B(y) = m_A(z) \sup m_B(w)$. Mode defuzzification will give the centroid for symmetric B and correlation-min inference but in general there will be a symmetric locus of mode points on each side of the centroid.

To minimize the local mean-squared error of the approximation the centroid of B should lie at the centroid $c(x)$ of the function f on the subinterval:

$$c(x) = \frac{1}{a + b} \int_{x-a}^{x+b} f(w)\, dw. \tag{6}$$

The centroid minimizes the mean-squared error of approximation[7] on the subinterval since it equals the conditional expectation evaluated at the subinterval:

$$c(x) = E[f \,|\, [x - a, x + b]] \tag{7}$$

for f restricted to the subinterval. [Eq. (7) also follows 7 by differentiating (2) with respect to c if $F = c$ on the subinterval.] Then we can view[1] the fuzzy set B as a random set or locus of two-point conditional probability densities. Then $m_B(y) = p(y \in B | Y = y)$ and $m_{B^c}(y) = p(y \notin B | Y = y)$.

The mean value theorem (MVT) of calculus[4] and (6) imply that $c(x) = f(z)$ for at least one point z in the subinterval. The MVT states that if f is continuous on an interval $[d, e]$ and differentiable on its interior (d, e), then there is at least one point z in (d, e) such that

$$f'(z) = \frac{f(e) - f(d)}{e - d} \tag{8}$$

where $f' = \frac{df}{dz}$. The integral in (6) gives $c(x)\ (a + b) = F(x + b) - F(x - a)$. So $c(x) = F'(z) = f(z)$ for some z in the subinterval since $a + b = (x + b) - (x - a)$.

The centroid line $c(x)$ cuts f in just one place if $f' > 0$ *or* $f' < 0$ and thus if f is monotone on the subinterval. Else it cuts f for two or more distinct arguments e and d in the interval. So $f(e) = f(d)$ since the centroid line has zero slope, and (8) becomes Rolle's theorem and gives $f'(z) = 0$ for some z in (e, d) and thus for some z in $(x - a, x + b)$. So f has an extremum in the subinterval and $m_A(z) > 0$. We can always widen B and keep the same centroid value B to make the rule patch $A \times B$ cover the extremum value $f(z)$ of the graph of the function. This use of the MVT is the key idea in the proof in the next section.

III. OPTIMAL FUNCTION APPROXIMATION WITH LONE FUZZY RULES: PATCH THE BUMPS

Where do we put the lone rule patch $A \times B$ to minimize the squared error E in (1)? We have to pick the best subinterval $I(x) = [x - a, x + b]$ of $[u, v]$. We slide the base of fuzzy set A across the interval $[u, v]$ and bring it to rest at the argument $\hat{x}$ that minimizes E.

Equations (5)–(7) imply that $F(z) = c(x)$ for z in $I(x)$ and $F(z) = 0$ for z outside of $I(x)$. So each x in $[u, v]$ defines a unique fuzzy system F_x. Each F_x uses the centroid or local mean-squared optimum. We need to find a mean-squared optimum $F_{\hat{x}}$ in the indexed family of fuzzy systems $\{F_x\}_{x \in X}$.

The total squared-error in (1) now depends on x and we write it as the function $E(x)$. It depends on each one-rule fuzzy system F_x. Each F_x is a rectangular pulse and that gives $E(x)$ as a sum of three integrals that depend on x:

$$E(x) = \int_u^v (f(w) - F_x(w))^2\, dw \tag{9}$$

$$= \int_u^{x-a} f^2(w)\, dw + \int_{x-a}^{x+b} (f(w) - c(x))^2\, dw + \int_{x+b}^{v} f^2(w)\, dw \tag{10}$$

The optimal fuzzy system $F_{\hat{x}}$ zeroes the derivative of E:

$$0 = \frac{dE(\hat{x})}{dx} \tag{11}$$

$$= (f^2(x-a) - f^2(x+b)) + \frac{d}{dx}\int_{x-a}^{x+b} (f(w) - c(x))^2\, dw \tag{12}$$

$$\begin{aligned} &= (f^2(x-a) - f^2(x+b)) + (f(x+b) - c(x))^2 - (f(x-a) - c(x))^2 \\ &\qquad - 2\int_{x-a}^{x+b} (f(w) - c(x))\frac{dc}{dx}\, dw \end{aligned} \tag{13}$$

by Leibniz's rule for differentiating under an integral sign

$$= (f^2(x-a) - f^2(x+b)) + (f(x+b) - c(x))^2 - (f(x-\mathrm{a}) - c(x))^2 \tag{14}$$

since the integral in (13) evaluates to zero:

$$\int_{x-a}^{x+b} (f(w) - c(x))\frac{dc}{dx}\, dw = \frac{dc}{dx}\int_{x-a}^{x+b} f(w)\, dw - c(x)\frac{dc}{dx}\int_{x-a}^{x+b} dw \tag{15}$$

$$\begin{aligned} &= (f(x+b) - f(x-a))\left[\frac{F(x+b) - F(x-a)}{a+b}\right] \\ &\qquad - c(x)\frac{f(x+b) - f(x-a)}{a+b}(a+b) \end{aligned} \tag{16}$$

$$= (f(x+b) - f(x-a))\,(c(x) - c(x)) = 0. \tag{17}$$

Then

$$0 = \frac{dE(\hat{x})}{dx} = 2c(x)\,[f(x+b) - f(x-a)] \tag{18}$$

In general the case $c(\hat{x}) = 0$ also leads to $f(\hat{x} + b) - f(\hat{x} - a) = 0$. (Monotone functions f have only endpoint extrema on $[u, v]$ and the case $c(x) = 0 = (f(x + b) - f(x - a)$ gives an inflection point.)

So the error is minimized when $f(x + b) = f(x - a)$. Then the mean value theorem implies that there is some z in $(x - a, x + b)$ such that $f'(z) = 0$. So the extremal value z belongs to fuzzy set A to some nonzero degree. Then the rule patch $A \times B$ covers the extremum point $(z, f(z))$ for a wide enough fuzzy set B: $m_{A\times B}(z, f(z)) > 0$. This optimal rule patches the bump $(z, f(z))$.

IV. OPTIMAL ADDITIVE FUZZY SYSTEMS WITH MANY RULES

An additive fuzzy system can have m rules $(\mathrm{A}_1, B_1), \ldots, (A_m, B_m)$. The *ith* if-part fuzzy set A_i covers a subinterval $[x_i - a_i, x_i + b_i]$. The *ith* then-part fuzzy set B_i has centroid c_i that equals the local centroid of $f(x)$ over $[x_i - a_i, x_i + b_i]$. The vector $\mathbf{x} = (x_1, \ldots, x_m)$ picks a fuzzy system $F_{\mathbf{x}}$. This gives an approximation error $E(\mathbf{x})$ in (9).

In general the subintervals overlap. Then each input z "fires" more than one rule or belongs to more than one set A_i to nonzero degree. The disjoint case gives lone rules and the analysis proceeds as above for a lone rule.

The number of rules m can differ from the number of extrema. In practice m is larger. Rule patches can interpolate between the flat extremal rule patches. We show that the fuzzy set A_i should "peak" above x_i. The optimal fuzzy system should equal the centroid of B_i when the input is $x_{i:}$ $F_{\hat{\mathbf{x}}}(x_i) = c_i$. In practice this means contiguous fuzzy sets should not overlap too much.

An additive fuzzy system[6] $F_{\mathbf{x}}$ takes the centroid of the sum of scaled then-part fuzzy sets:

$$F(z) = \text{Centroid}(m_{A_1}(z)B_1 + \ . \ . \ . + m_{A_m}(z)B_m) \tag{19}$$

$$= \frac{\sum_{j=1}^{m} \text{Volume}\,(m_{A_j}(z)B_j)\ \text{Centroid}(B_j)}{\sum_{j=1}^{m} \text{Volume}\,(m_{A_j}(z)B_j)} = \frac{\sum_{j=1}^{m} V_j m_{A_j}(z) c_j}{\sum_{j=1}^{m} V_j m_{A_j}(z)} \tag{20}$$

where the area or volume $V_j = \int_Y m_{B_j}(y)\,dy$. So (20) reduces to (5) if z belongs only to A_i to nonzero degree.

To find the $\hat{\mathbf{x}}$ that minimizes $E(\mathbf{x})$ we take the gradient of E with respect to $\mathbf{x}$ and set it equal to the null vector and solve. Each integrand now involves $F(w)$ instead of just the constant $F(x_i)$ as in (10) since the if-part sets overlap:

$$\begin{aligned} 0 = \frac{\partial E}{\partial x_i} &= \frac{\partial}{\partial x_i}\int_u^{x_i - a_i} (f(w) - F(w))^2\,dw \\ &+ \frac{\partial}{\partial x_i}\int_{x_i - a_i}^{x_i + b_i} (f(w) - F(w))^2\,dw + \frac{\partial}{\partial x_i}\int_{x_i + b_i}^{v} (f(w) - F(w))^2\,dw \end{aligned} \tag{21}$$

$$= 2\int_{x_i - a_i}^{x_i + b_i} (f(w) - F(w))\frac{\partial F(w)}{\partial x_i}\,dw. \tag{22}$$

Then $\frac{\partial E}{\partial x_i} = 0$ when $f = F$ or when

$$0 = \frac{\partial F}{\partial x_i} = V_i \frac{dm_{A_i}}{dx}\left[c_i \sum_{j=1}^{m} V_j m_{A_j} - \sum_{j=1}^{m} V_j m_{A_j} c_j\right] \tag{23}$$

So the approximation error is minimized when

$$\frac{dm_{A_i}}{dx_i} = 0 \tag{24}$$

or when

$$c_i = \frac{\sum_{j=1}^{m} V_j m_{A_j} c_j}{\sum_{j=1}^{m} V_j m_{A_j}} \tag{25}$$

Then (20) and (25) imply that $F_{\hat{\mathbf{x}}}(\hat{x}_i) = c_i$. A neural or other adaptive system may move the peak of A_i away from x_i as long as (25) still holds.

These optimality results let us apply the lone-rule result for minimizing an error function with isolated extrema. First we try to find the bumps $f'(\hat{z}_i) = 0$. Then we *center* if-part sets A_i at these points: $\hat{x}_i = \hat{z}_i$. Then we add more if-part sets A_i that both maintain (24) and (25) and that patch the bumps of the new *residual* error function.

References

1. B. Kosko, "Fuzzy systems as universal approximators," *Computers,* **43**(11), 1329–1333 (November 1994); an earlier version appears in the *Proceedings for the 1st IEEE International Conference on Fuzzy Systems (IEEE FUZZ-92),* March 1992, pp. 1153–1162.
2. J.A. Dickerson and B. Kosko, "Fuzzy function approximation with supervised ellipsoidal learning," *World Congress on Neural Networks (WCNN-93),* **2,** 9–17 (July 1993).
3. P.J. Pacini and B. Kosko, "Adaptive fuzzy system for target tracking," *Intell. Syst. Engi.,* **1**(1), 3–21 (Fall 1992).
4. E. Kreyszig, *Advanced Engineering Mathematics,* 6th ed. J Wiley, New York, 1988.
5. J.R. Rice, *The Approximation of Functions,* Addison-Wesley, Reading, MA, 1964.
6. B. Kosko, *Neural Networks and Fuzzy Systems,* Prentice-Hall, Englewood Cliffs, NJ, 1991.
7. A. Papoulis, *Probability, Random Variables, and Stochastic Processes,* 2nd ed. McGraw-Hill, New York, 1984.